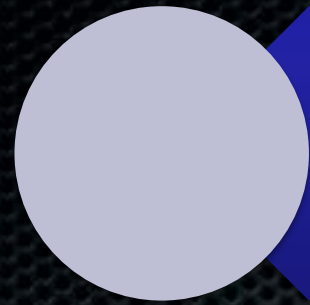


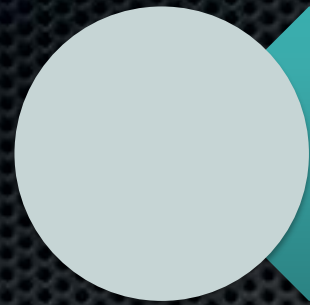
Death by 32 Bits



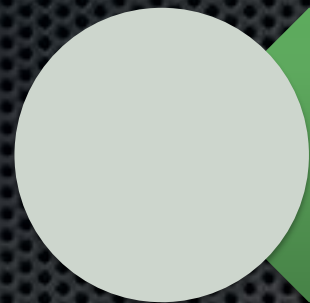
4,294,967,296



Human Population



IPv4 Addresses



Fast Networks



Cheap Memory



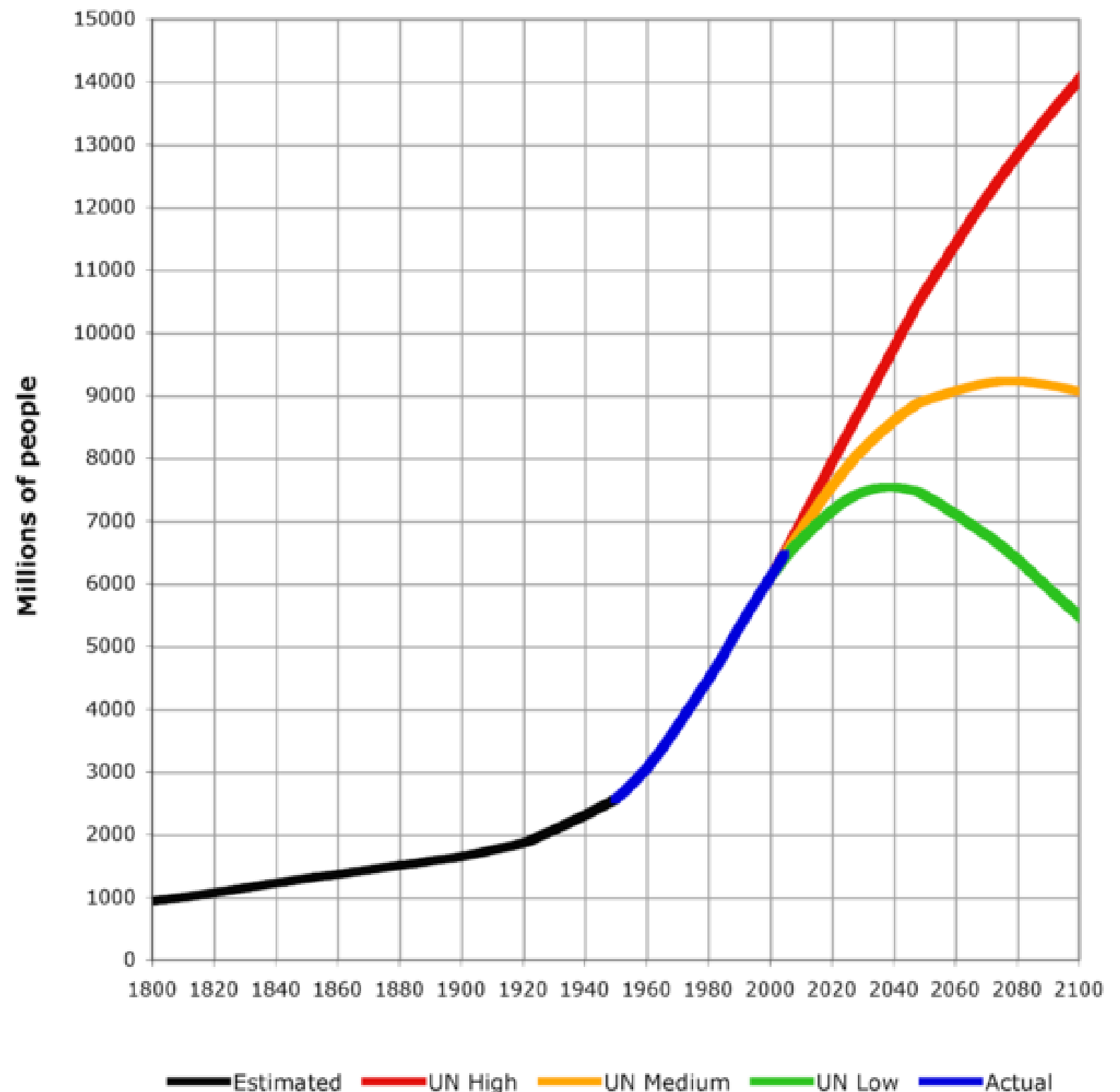
32-bit Processors

World Population
6 billion+

China
1.3 billion+

India
1.1 billion+

USA
305 million+



Internet Usage

China

22.48%

USA

72.35%

Growth Rates

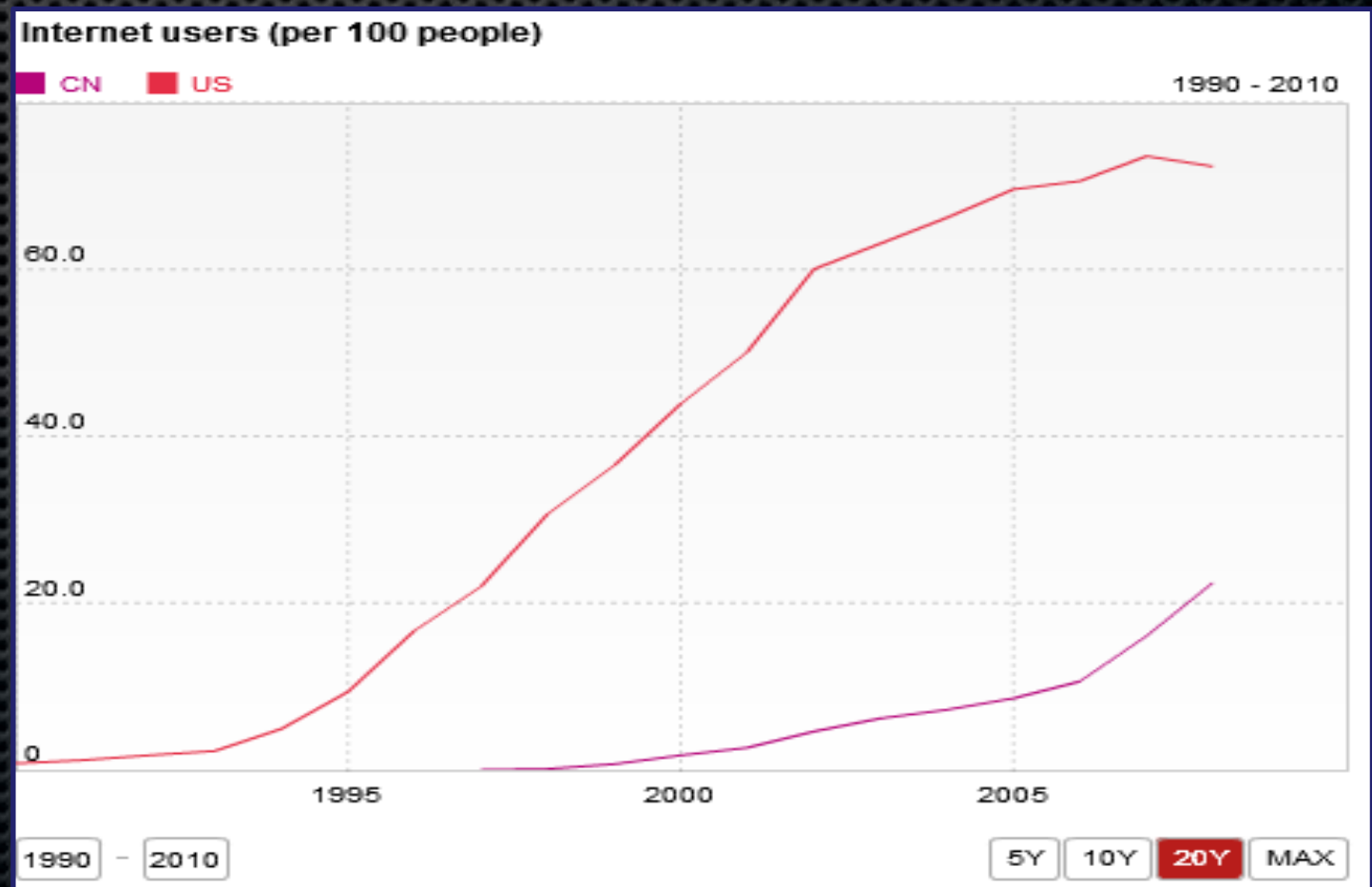
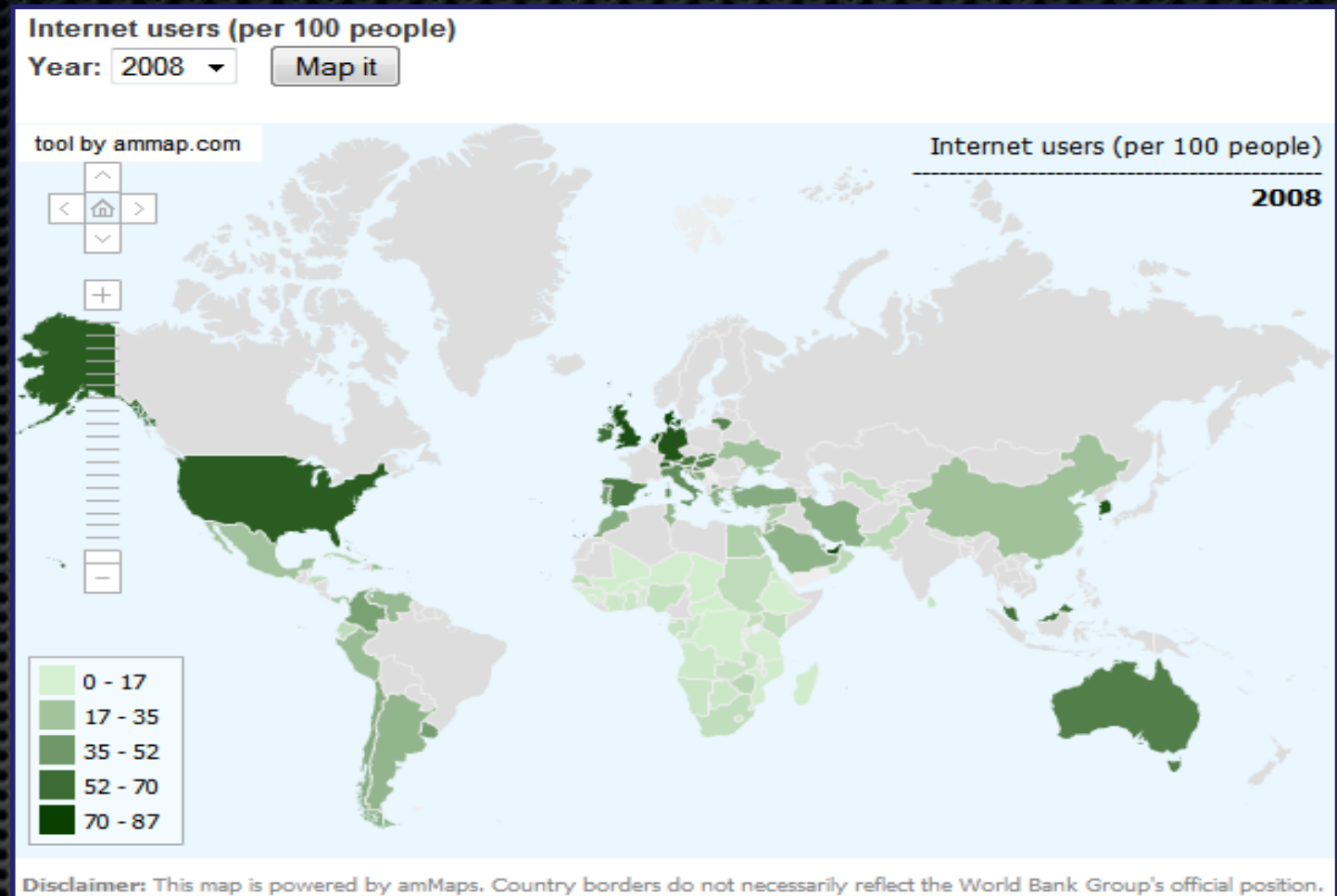
USA

72% 12 years ago

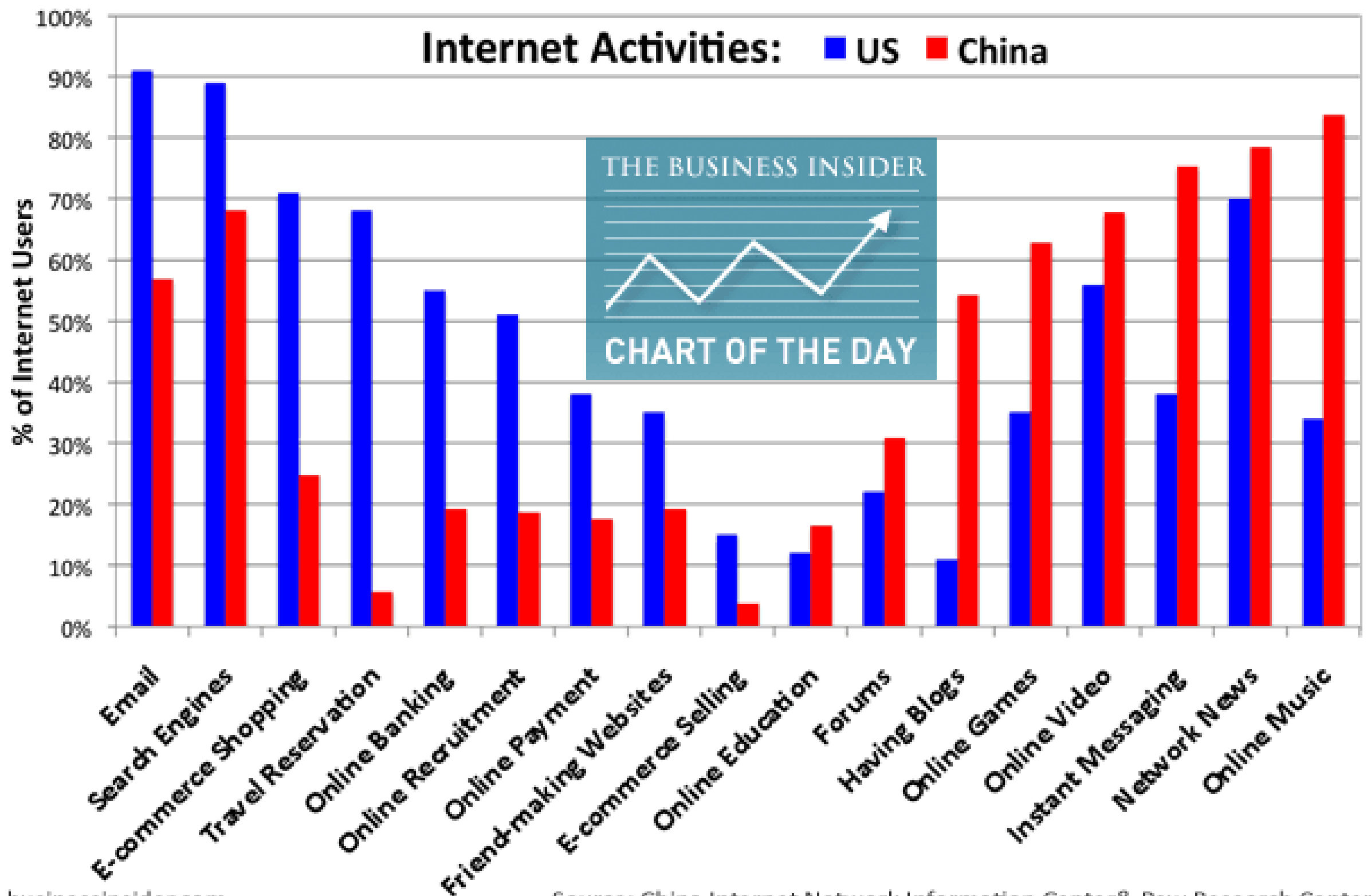
Flat since 2007

China

50% by 2012?



Internet Usage - USA vs China



Internet Population

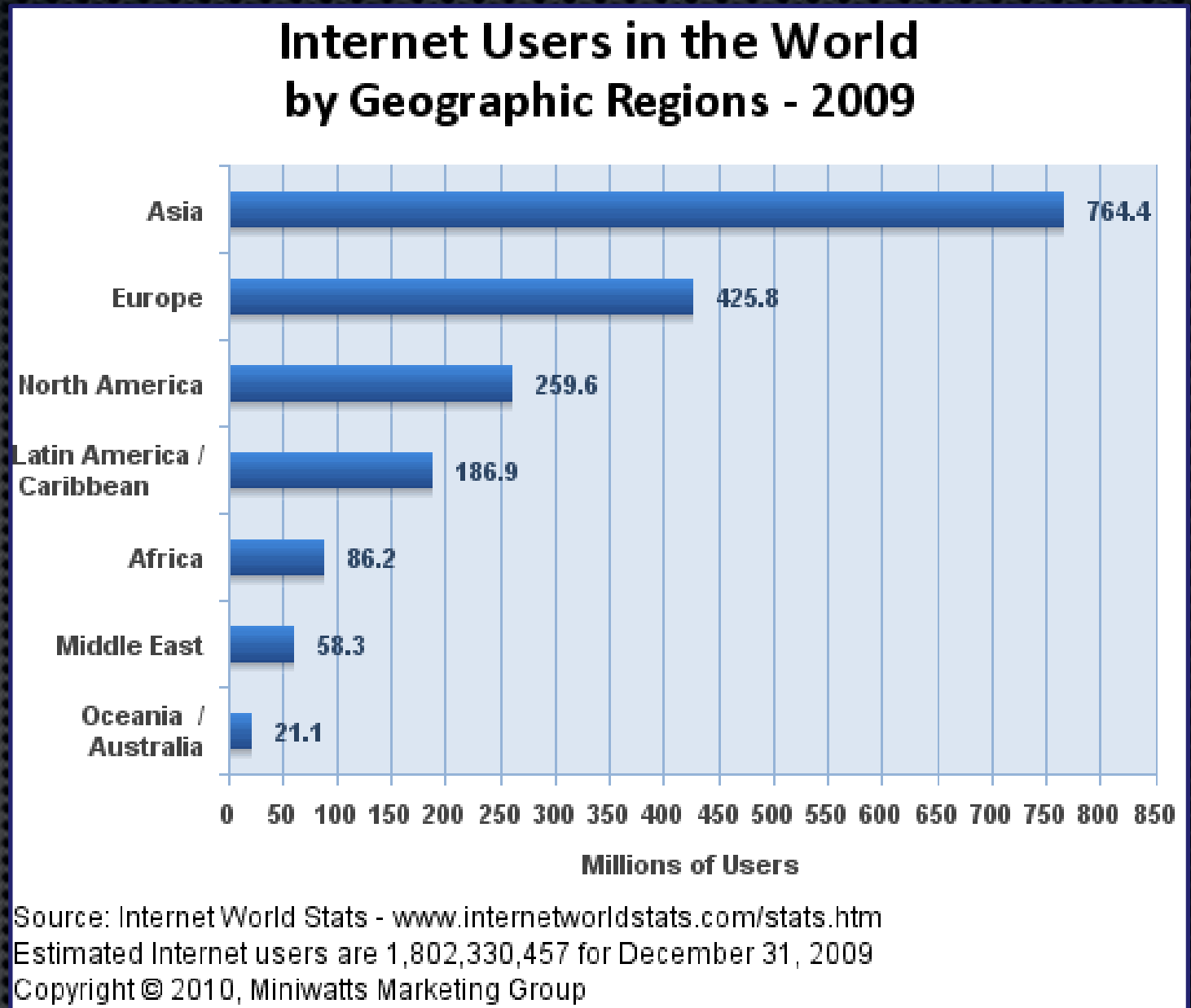
1.8 billion+

China

300 million+

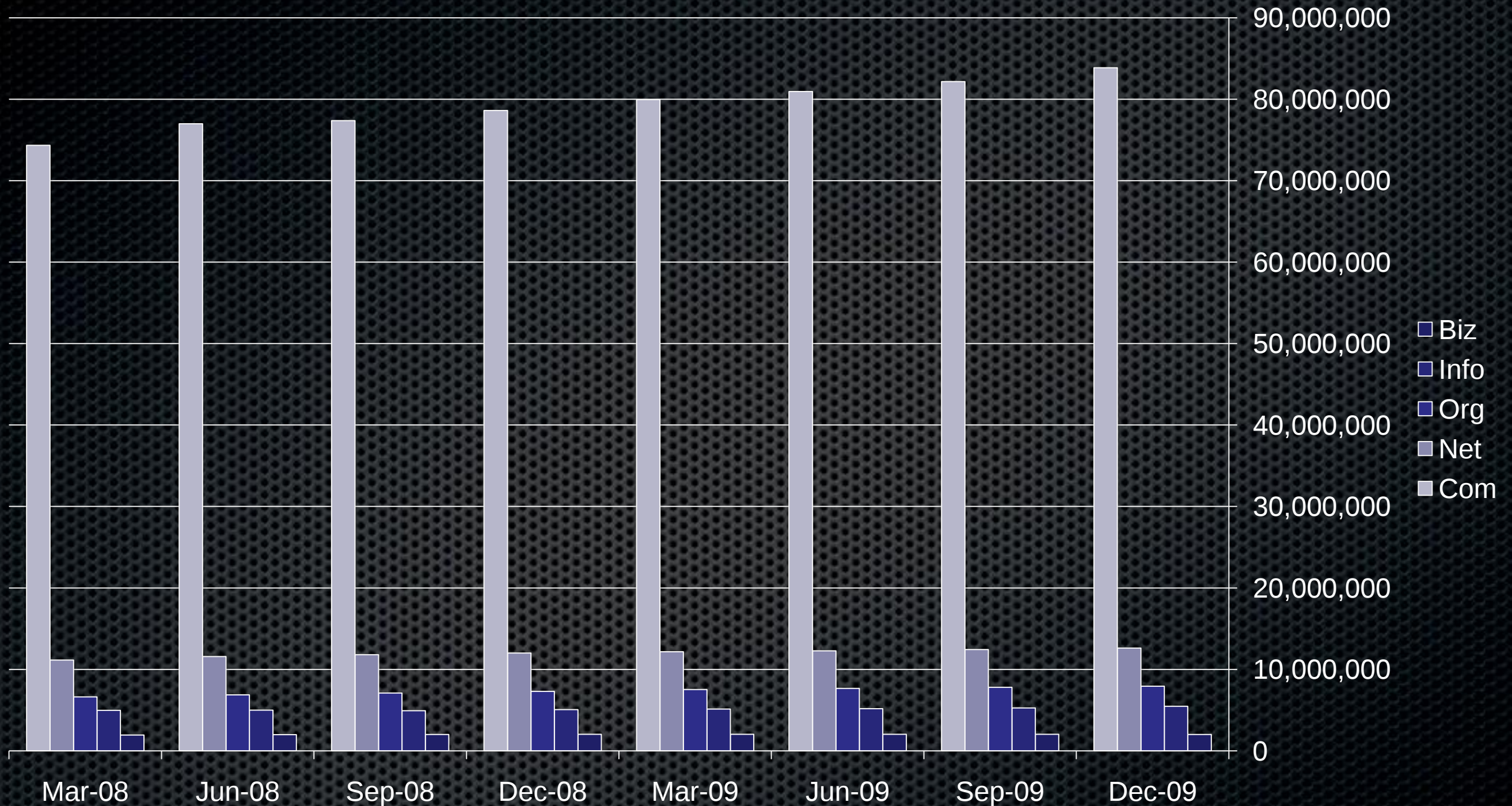
USA

200 million+



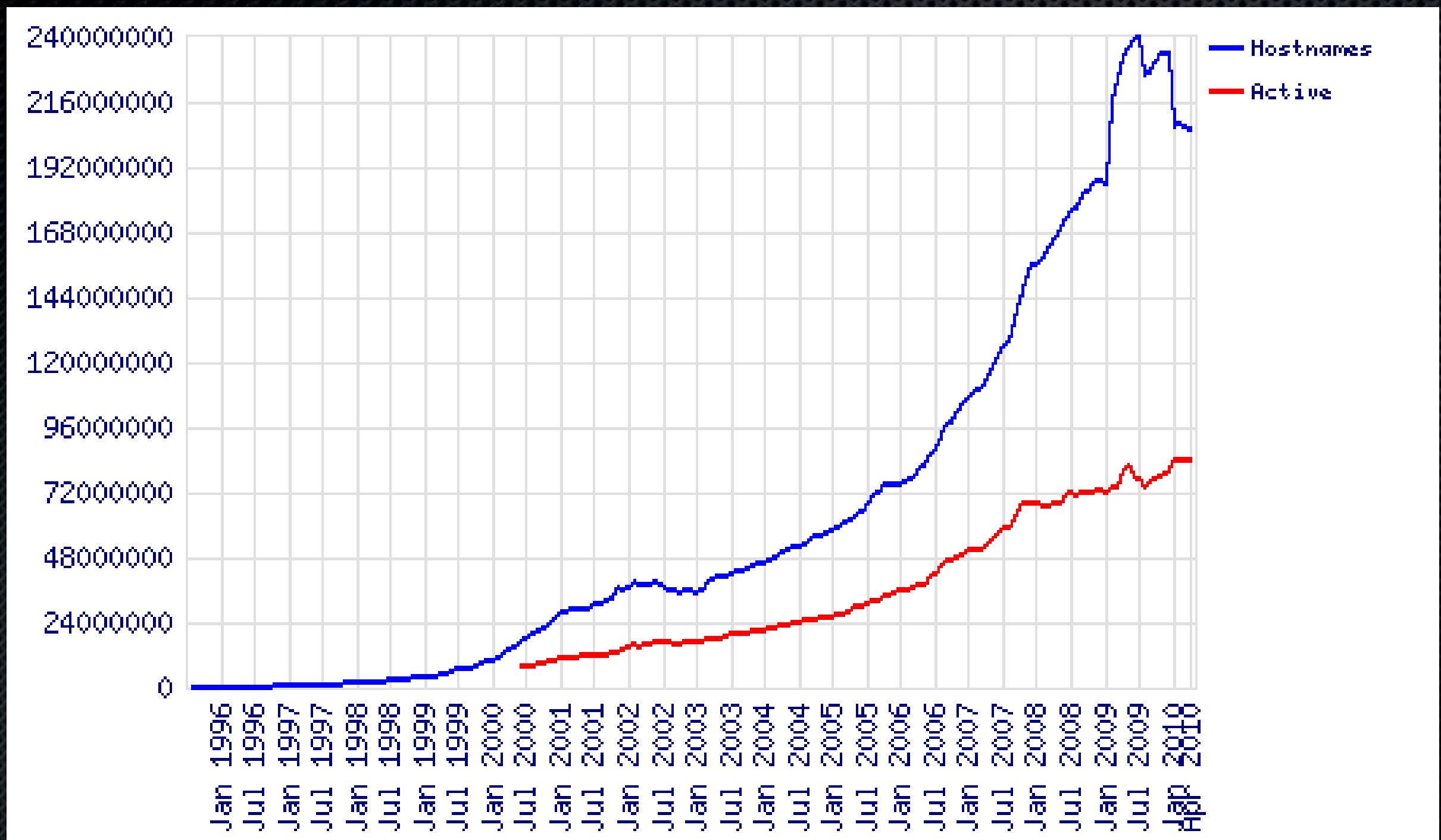
1.8 billion is **42%** of the 32-bit max

Domain Names: 2008 to 2009



84 million **registered** .coms

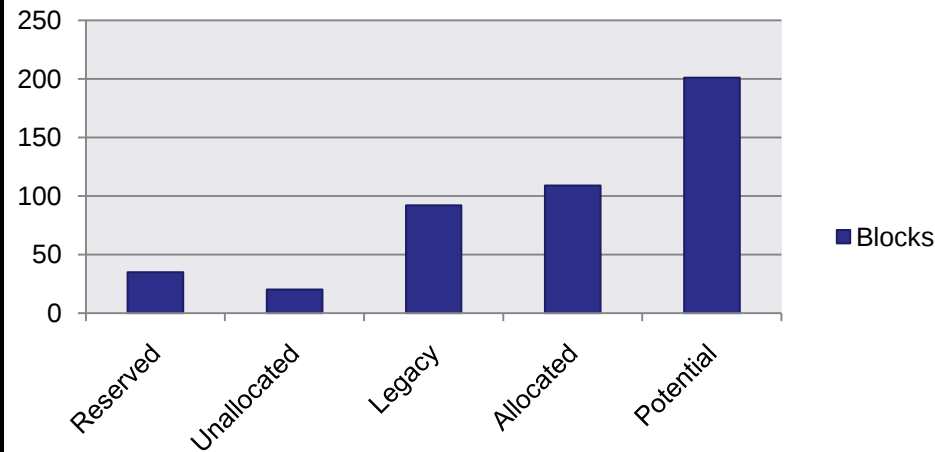
Active Sites: 1996 to 2010 (Netcraft)



84 million **active** web sites

Allocated IPv4 Address Space

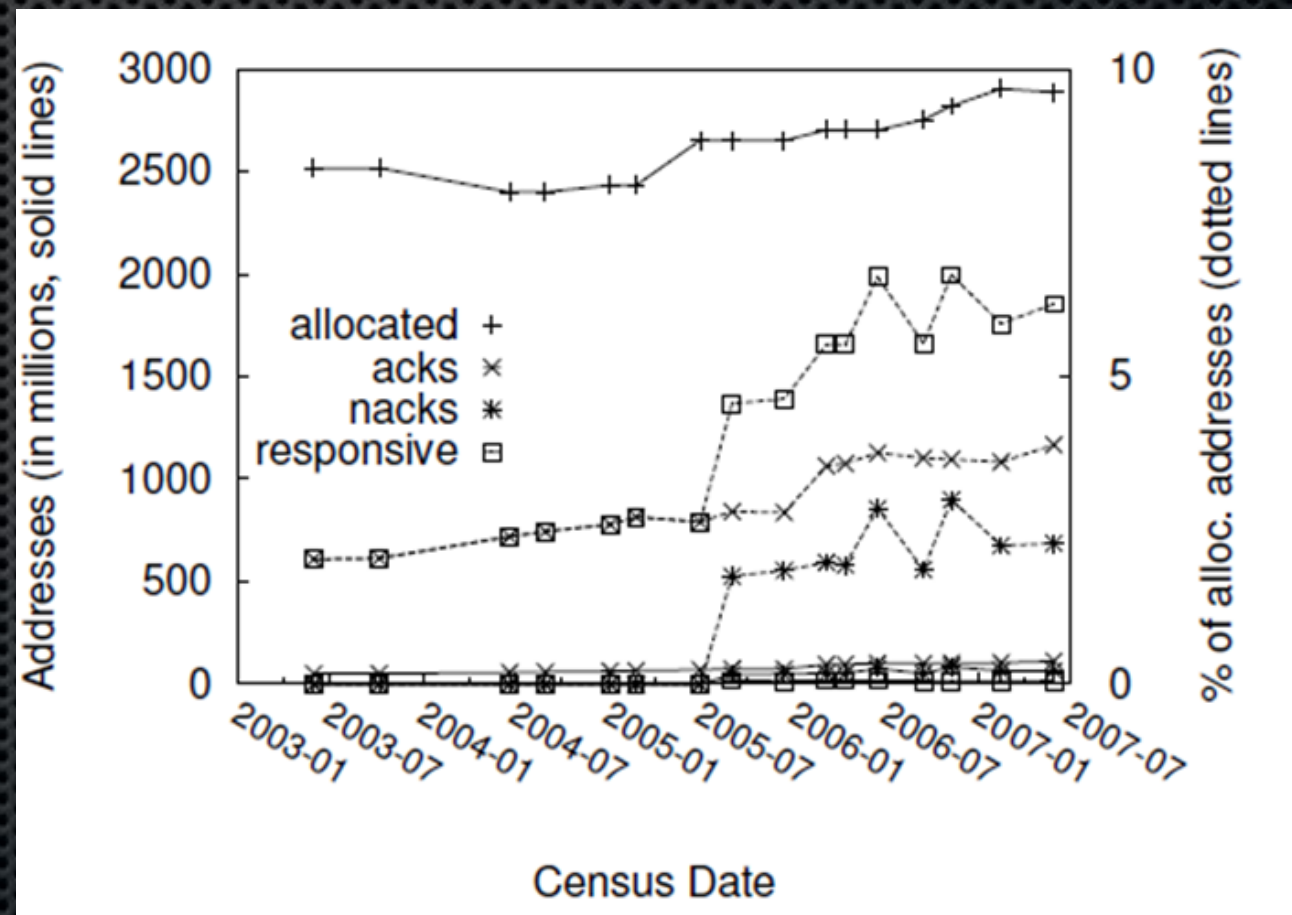
IPv4 Address Blocks (/8)



Total number of IPv4 addresses:

2 ³² :	4294967296	4294.97 million
Class D+E:	536870912 -	536.87 million -
Nets 0 and 127:	33554432 -	33.55 million -
RFC 1918:	17891328 -	17.89 million -
-----		-----
Usable:	3706650624	3706.65 million

IPv4 Addresses
3.70b possible
3.37b allocated
334m available
~1.7b active*



Source: <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.txt>

Source: <http://www.bgpexpert.com/addressespercountry.php>

Source: <http://www.isi.edu/~johnh/PAPERS/Heidemann08a.pdf>

Population vs Domains vs IP Addresses

Approximate ratios

- 1** internet user per 3.72 humans
- 1** user per active IP address
- 9** users per registered hostname
- 17** US residents per 100 users
- 21** users per registered .com
- 21** users per active web site

IP address ratios

- 86%** of the IPv4 space is usable
- 91%** of usable space is allocated
- 50%** of this space is active

Packet Transmission Speed

A 1000 byte packet, once per second

$$1000 \text{ bytes} * 8 \text{ bits} = 8 \text{ kbps}$$

A 40 byte packet, once per second

$$40 \text{ bytes} * 8 \text{ bits} = 0.32 \text{ kbps}$$

A 100m ethernet network card

$$1514 \text{ bytes} * 8 \text{ bits} = 12.12 \text{ kb}$$

$$1514 \text{ bytes} * 8246/\text{sec} = 100 \text{ Mbps}$$

$$40 \text{ bytes} * 312500/\text{sec} = 100 \text{ Mbps}$$

Reality is more complicated (IPG, software)

Decent server can send about 50k pps

Bandwidth required is 400k/byte

Network Bandwidth vs IPv4 Space

Single-request TCP exploit (conn + send)

3.5 days = 3.37b * 4 @ 50k pps

Single-packet exploit to ALL allocated IPs

19 hours = 3.37b @ 50k pps

Single-packet exploit vs US

8.34 hours = 1.50b @ 50k pps

Single-packet exploit vs China

1.37 hours = 247m @ 50k pps

Single-packet exploit vs Russia

10.3 minutes = 31m @ 50k pps

Network Bandwidth vs Clouds

Bandwidth is relatively cheap

Small packets = low bandwidth

Billing is based on "transfers"

Clouds makes blocking the source hard

Get a **new IP** anytime you like

Handy for **penetration tests**

Clouds make internet-wide attacks easy

10 servers = Russia in 60 seconds

Cost = ~\$50.00 USD

IPv6 – 128 bits of fun

Network ranges become “unscannable”

- Hosts are allocated a /64 each

Finding systems becomes the hard part

- Local networks are discoverable
- Remote networks depends on DNS

Legacy software rarely binds to IPv6

- Fewer extra services running

Still some downsides

- Not all firewalls block IPv6 correctly
- Easy to hide remote rogue systems
- Hosts are IPv6 ready, users are not

System Memory Pricing

RAM is cheap

\$23.00 for 1Gb (DDR3 @ 1333Mhz)

\$0.02 per megabyte

Netbooks ship with 1G or 2g

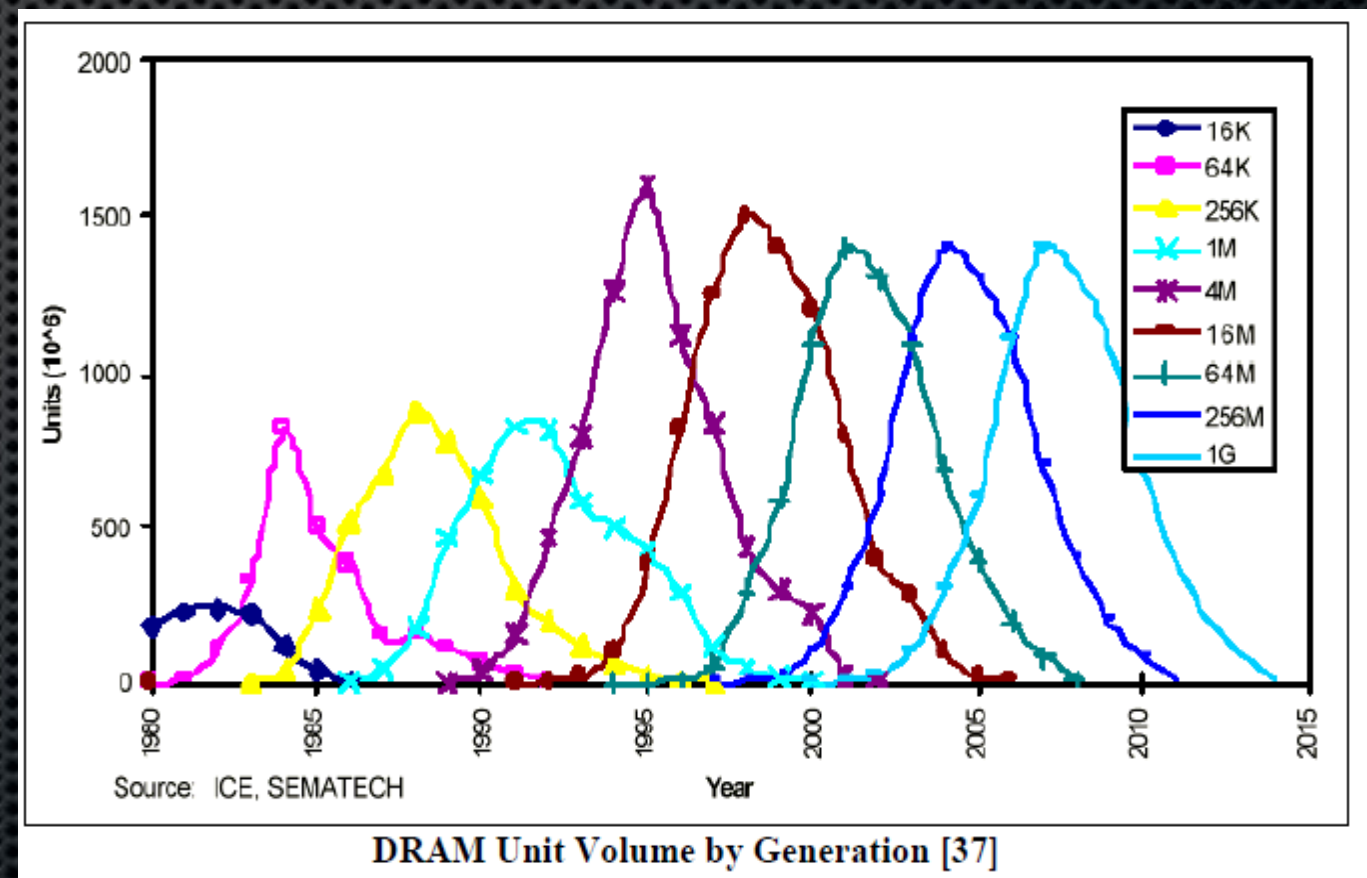
Video cards "average" 512M

Supply drives price

6 years to peak

Old RAM costs more

Based on supply



Source: http://www.pricewatch.com/system_memory/

Source: http://www.tezzaron.com/about/papers/dram_pricing.pdf

Source: <http://store.steampowered.com/hwsurvey/>

System Memory Availability

Cheap RAM increases software requirements

- Windows 2000 **32Mb** minimum
- Windows 7 **1024Mb** minimum
- Office 2000 **8Mb** minimum (+OS)
- Office 2010 **256Mb** minimum (+OS)

Gamers (as usual) are a good indicator of trend

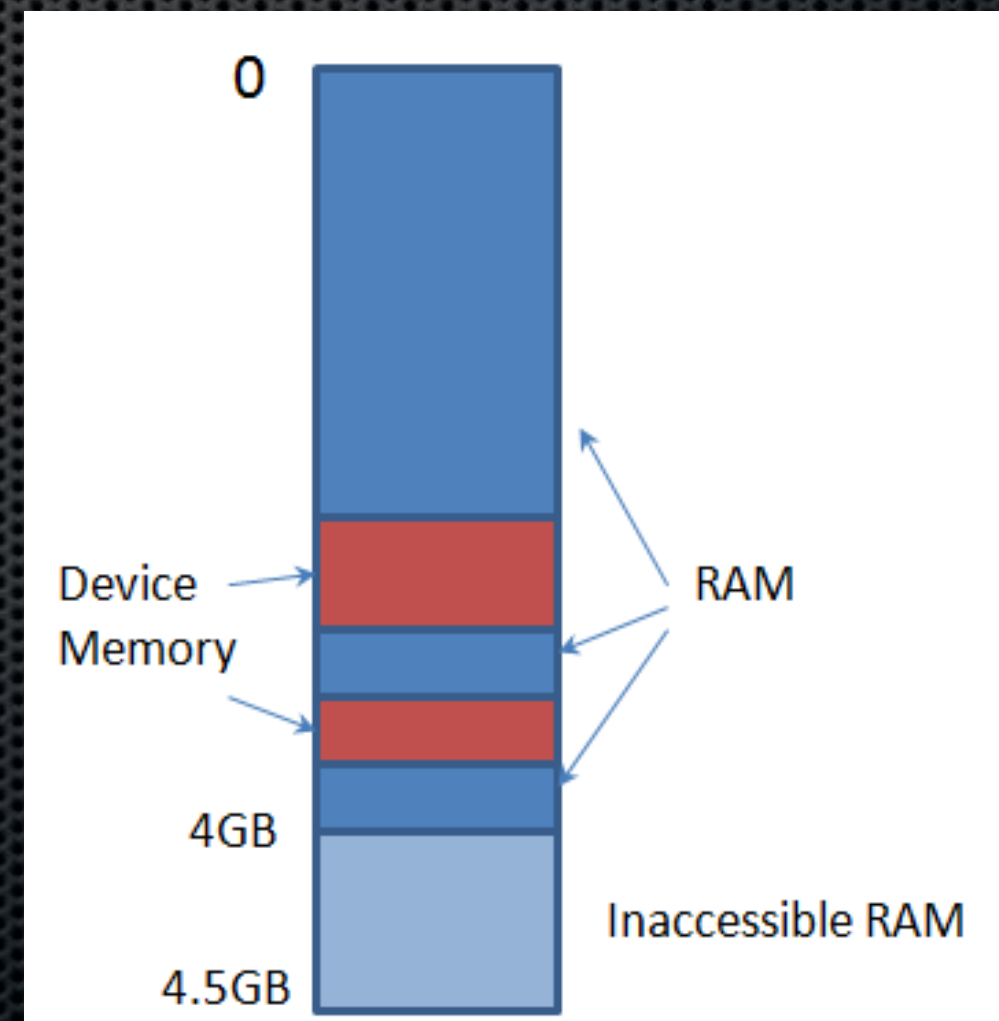
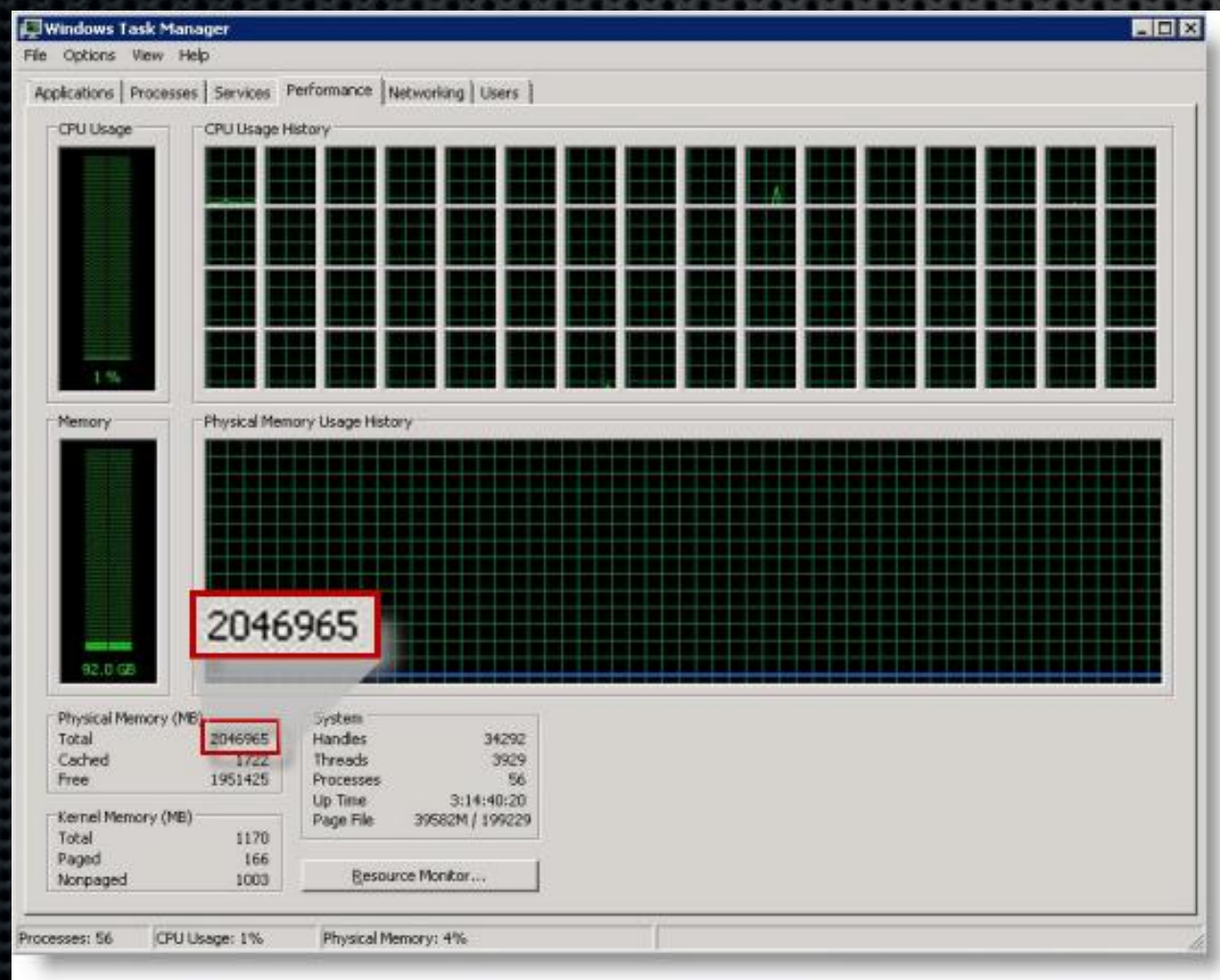
- 84%** have 2Gb or more
- 27%** have 4Gb or more
- 4%** have less than 1G

Less than 512 MB	(-0.13%)	0.66%
512 Mb to 999 MB	(-0.31%)	3.88%
1 GB	(-0.33%)	10.93%
2 GB	(-0.69%)	28.82%
3 GB	(0.00%)	28.09%
4 GB	(+0.76%)	18.19%
5 GB and higher	(+0.70%)	9.43%

System Memory vs 32-bit Processors

32-bit CPUs can only address 32-bits of memory

- Virtual memory must also include device I/O
- PAE and other tricks help, but are not efficient
- Real maximum is between **2.0Gb** and **3.5Gb**



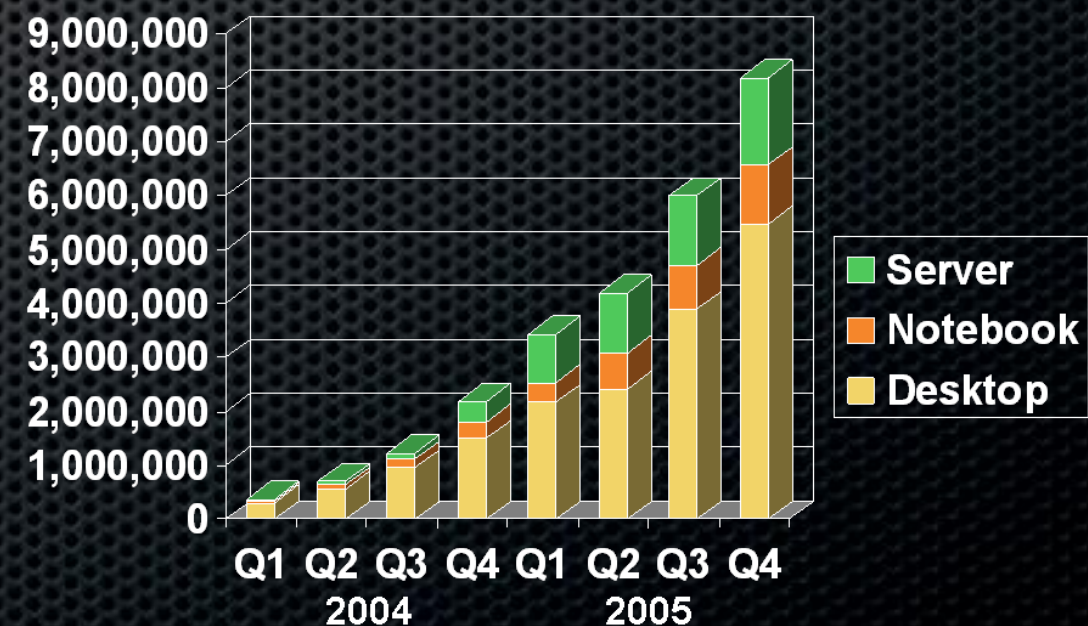
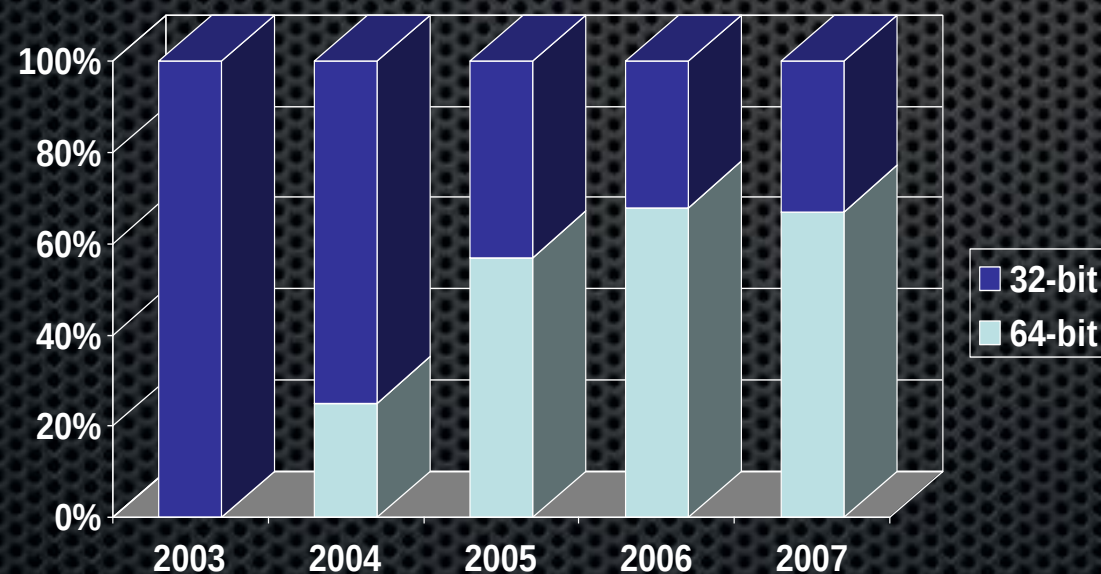
32-bit vs 64-bit Penetration

We turn to the Gamers for trends

33% run 64-bit Windows
28% run 32-bit Vista / 7
54% of Vista / 7 are 64-bit!

Windows XP 32 bit	(-1.72%)	38.61%
Windows 7 64 bit	(+1.43%)	24.42%
Windows Vista 32 bit	(-0.19%)	16.69%
Windows 7	(+0.33%)	11.25%
Windows Vista 64 bit	(+0.15%)	7.75%
Windows XP 64 bit	(+0.02%)	0.62%
Windows 2003 64 bit	(-0.03%)	0.44%
Windows 2000	(+0.10%)	0.10%
Other	(-0.08%)	0.12%

Great stats from Microsoft WinHEC 2006



Source: <http://store.steampowered.com/hwsurvey/>

Source: http://download.microsoft.com/download/5/b/9/5b97017b-e28a-4bae-ba48-174cf47d23cd/BUS080_WH06.ppt

32-bit Exploit Mitigations

Newer operating systems try to block exploits

- Prevent execution of data: **DEP + NX**
- Limit predictability of memory: **ASLR**
- Limit exception handlers: **/SafeSEH**
- Prevent return address overwrites: **/GS**

Newer techniques bypass most if not all

- Bypass **/GS** with smashed exception handlers
- Sometimes bypass **/SafeSEH** with **VEH**
- Bypass **DEP** with Return-Oriented-Programming (**ROP**)
- Bypass **ASLR** with heap spraying or brute forcing

Security mitigations are limited by the 32-bit platform

32-bit Integers

x86 integers indicate sign in the high bit

- **0x00000001** = 1 signed or 1 unsigned
- **0xFFFFFFFF** = -1 signed or 4,294,967,296 unsigned
- **0x7FFFFFFF** = 2,147,483,647
- **0x80000000** = -2,147,483,648

Even smart coders didn't account for huge input

```
int i = strlen(input); // casting bug  
if (i < MAX_LEN)  
    badness();
```

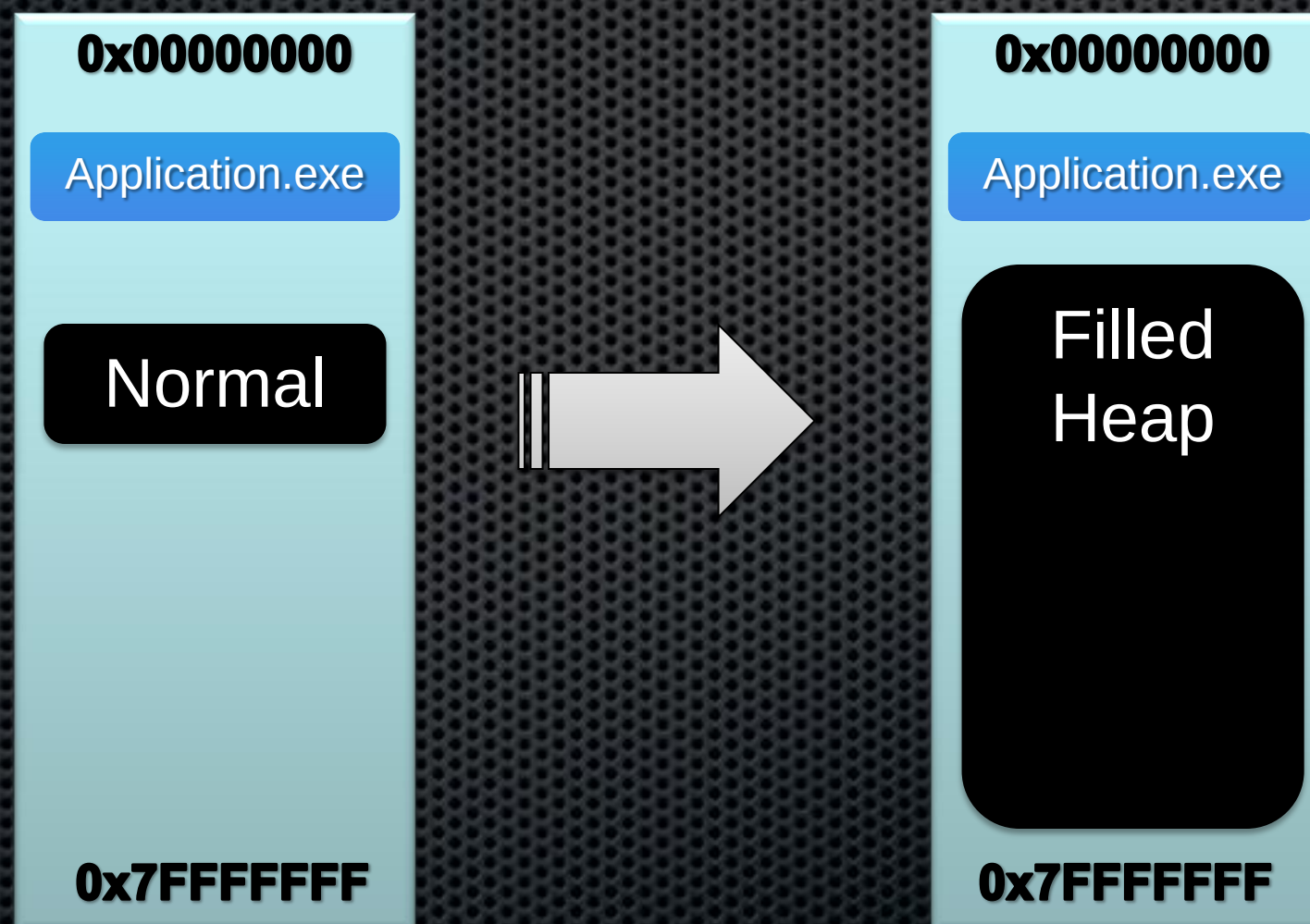
Solutions for legacy code?

- Set process memory limits to under 2G
- Force migration to 64-bit platforms

32-bit Memory Prediction

The 32-bit virtual memory space is relatively tiny

- Attacker supplied files or scripts negate ASLR
- Most client-side applications are vulnerable
- Address prediction leads to DEP bypass



32-bit Attacker Memory Control

The user process is normally limited to 2Gb

- Transferring 2Gb of data is not feasible (yet)
- Client-side code can easily allocate memory
 - Javascript, Java, Flash, .NET, etc

Trivial to do without client-side scripting

- Builtin protocol compression (gzip, deflate)
- Compressed containers (docx, odt, zip, ole)
- Compressed graphics and sound (mp3, png)

Often possible against server-side applications

- Protocol compression works as well (SSL)
- XDR and NDR encoding control allocations
- HTTP Content-Length and File Uploads

32-bit Memory Control via Graphics

24-bit graphics are ubiquitous

- Pixels stored as one byte for Red, Blue, and Green
- 32-bit graphics include one byte alpha channel
- Allows for 16.7 million colors per pixel plus alpha
- Memory allocation determined by dimensions

Examples

- 1 x 1 white block with no transparency

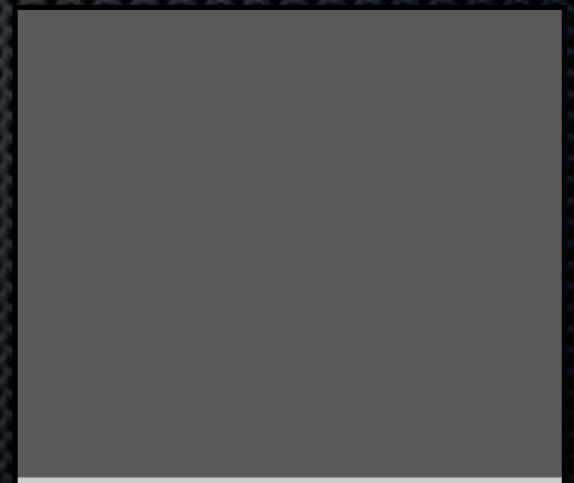
FF FF FF 00

- 32 x 32 white block with full transparency

FF FF FF FF x 1024 (4096 bytes)

- 16384 x 16384 image for x86 “debug trap”

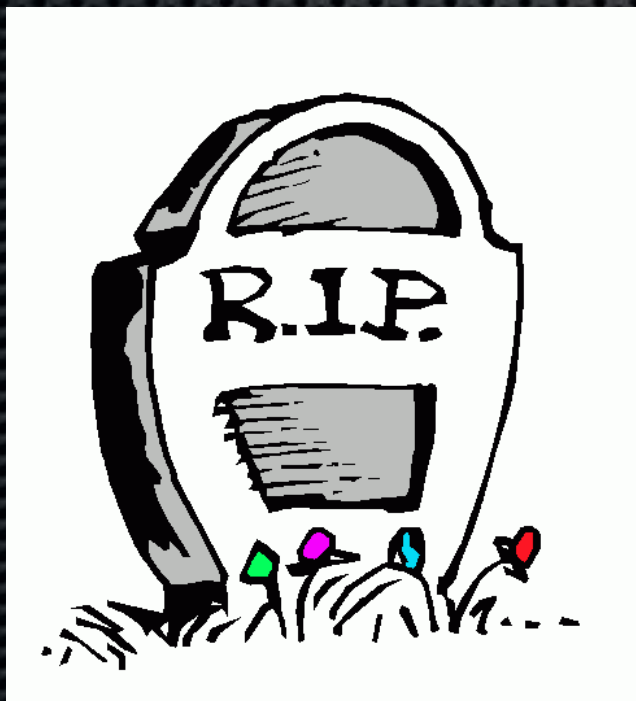
CC CC CC CC x 268435456 (1Gb+)



32-bit Application Security

Eulogy

- 32-bit app developers never expected 2 Gb of input
- Mitigation methods are limited by the platform
- Only so random a 32-bit value can become

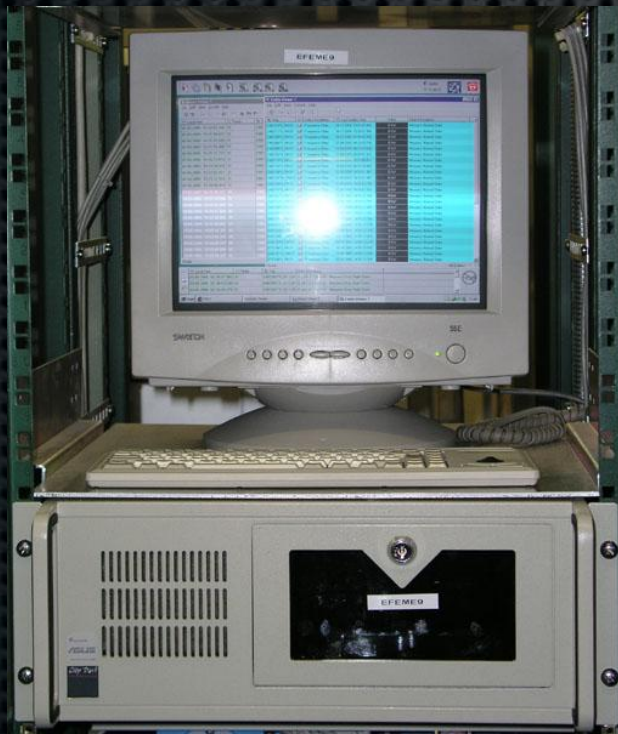


```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
              // guaranteed to be random.
}
```


32-bit Legacy

32-bit is here to stay

- 32-bit x86 is the “new” platform for SCADA gear
- x64 is backwards compatible with 32-bit x86
- Embedded CPUs are primarily 32-bit (ARM, MIPS)



64-bit Application Security

64-bit computing has numerous security benefits

- No need for software DEP, NX is built-in
- The stack is non-executable by default
- Randomization actually effective (48-bits)
- Better kernel protection in Windows
- ELF64 ABI mandates register passing

“This is the end of exploit development” - **<anonymous>**

64-bit Application Security

64-bit builds can actually be less secure

- Qmail on 64-bit is **trivially exploitable** (and unpatched)
- Problems when **64-bit pointers** meet **32-bit integers**
- Windows 64-bit still runs exploitable 32-bit apps
- Unexploitable 32-bit bugs become possible
- Return Oriented Programming (**ROP**) still possible

4,294,967,296

is a small number after all

HD Moore < hdm [at] metasploit.com >